FIRST SEMIANNUAL TECHNICAL REPORT

(15 October 1969 - 15 June 1970)

FOR THE PROJECT

"ANALYSIS AND OPTIMIZATION OF

STORE-AND-FORWARD COMPUTER NETWORKS"

Principal Investigator

and Project Manager:

HOWARD FRANK (516) 671 - 9583

ARPA Order No. 1523

Project No. 0D30

Contractor: Network Analysis Corporation

Contract No. DAHC15-70-C-0120

Effective Date: 15 October 1969

Expiration Date: 15 October 1970

Amount of Contract: $103,114.00

Sponsored by

Advanced Research Projects Agency

Department of Defense

ARPA Order No. 1523

68

SUMMARY

The ARPA Computer Network will provide communication paths between a set of computer centers distributed across the United States. The purpose of the network is to inexpensively and rapidly make available to all of the network's users, the special capabilities of each of the computer centers. The ARPA Contract with the Network Analysis Corporation involves the analysis and design of this network and a study of the properties of networks of this type.

The objectives of the project are:

(1) To develop computer programs which can determine economical data line locations and line capacities.

(2) To operate the programs in order to determine the appropriate data lines to be leased from AT&T, and the cost-throughput-time delay characteristics for store-and-forward networks such as the ARPA Network.

(3) To study the properties of large store-and-forward computer networks and to develop a specific example exhibiting the cost-throughput characteristics of a large network.

(4) To study the effect on network performance of alternate routing procedures and the amount of storage at each node.

Each network to be designed must satisfy a number of constraints. It must be reliable, it must be able to accommodate variations in traffic flow without significant degradation in performance, and it must be capable of efficient expansion when new nodes and links are added at a later date. Each design must have an average response time for short messages no greater than 0.2 seconds. The goal of the optimization is to satisfy all of the constraints with the least possible cost per bit of transmitted information.

Objectives (1) - (2) of the project have been completed and objective (3) will be completed shortly. An operational computer program has been developed. This program is capable of: (1) analyzing proposed network designs, and (2) finding economical combinations of lines which lead to highly efficient low cost network designs. The general design philosophy followed as well as the specific elements considered in the implementation of the program are described in the report. The computer program was used to determine the most economical lines which can be leased to satisfy the communication requirements of the ARPA Net-

work. It has also been used to study the relationships between traffic level, link capacities, and cost as a function of the number of nodes in the network. Extensive studies have been made for twelve, sixteen, eighteen, and twenty node networks where each node was a potential site for the ARPA Network. A number of the results of these studies are summarized in this report.

Highly efficient algorithms have been developed and programmed for the study of large computer networks. Methods for optimizing the design of centralized networks have been discovered. These methods, which are described here, are presently being used to design large decentralized hierarchal networks.

The problem of routing is under investigation and preliminary results will be reported shortly.

CONTENTS

Section	Page
INTRODUCTION	1
TOPOLOGICAL OPTIMIZATION	4
APPROACH	6
DESIGN CONSTRAINTS	11
COMPUTATIONAL RESULTS	22
CENTRALIZED NETWORKS	44
REFERENCES	62

INTRODUCTION

The ARPA Network will provide store-and-forward communication paths between a set of computer centers distributed across the continental United States. The message handling tasks at each node in the network is performed by a special purpose Interface Message Processor (IMP) located at each computer center. The centers will be interconnected through the IMPS by fully duplex telephone lines, of typically 50 kilobit/sec capacity.

When a message is ready for transmission, it will be broken up into a set of packets, each with appropriate header information. Each packet will independently make its way through the network to its destination. When a packet is transmitted between any pair of nodes, the transmitting IMP must receive a positive acknowledgment from the receiving IMP within a given interval of time. If this acknowledgment is not received, the packet will be retransmitted, either over the same or a different channel depending on the network routing doctrine being employed.

One of the design goals of the system is to achieve a response time of less than 0.2 seconds for short messages. A measure of the efficiency with which this criterion is met is the cost per bit of information transmitted through the network when

the total network traffic is at the level which yields 0.2 second average time delay. The goal of the network design is to achieve the required response time with the least possible cost per bit. The final network design is subject to a number of additional constraints. It must be reliable, it must have reasonably flexible capacity in order to accommodate variations in traffic flow without significant degradation in performance, and it must be neatly expandable so that additional nodes and links can be added at later dates. The sequence and allowable variations with which the nodes are added to the network must also be taken into account. At any stage in the evolution of the network, there must be at least one communication path between any pair of nodes that have already been activated. In order to achieve a reasonable level of reliability, the network must be designed so that at least two nodes and/or links must fail before the network becomes disconnected.

To plan the orderly growth of the network, it is necessary to predict the behavior of proposed network designs. To do this, traffic flows must be projected and network routing procedures specified. The time delay analysis problem has been studied by Kleinrock [1, 2] who considered several mathematical models of the ARPA Network. Kleinrock's comparison of his analysis with computer simulations indicates that network behavior can be

qualitatively predicted with reasonable confidence. However, additional study in this area is needed before all the significant parameters which describe the system can be incorporated into the model. For the present, it appears that a combination of analysis and simulation can best be applied to determine a specific network's behavior.

Even if a proposed network can be accurately analyzed, the most economical networks which satisfy all of the constraints are not easily found. This is because of the enormous number of combinations of links that can be used to connect a relatively small number of nodes. It is not possible to examine even a small fraction of the possible network topologies that might lead to economical designs. In fact, the direct enumeration of all such configurations for a twenty node network is beyond the capabilities of the most powerful present day computer.

TOPOLOGICAL OPTIMIZATION

As part of NAC's study of computer network design, a computer program was developed to find low cost topologies which satisfy the constraints on network time delay, reliability, congestion, and other performance parameters. This program is structured to allow the network designer to rapidly investigate the tradeoffs between average time delay per message, network cost, and other factors of interest.

The inputs to the program are:

1. Existing network configuration (i.e., lines and nodes already installed and ordered)
2. Estimated traffic between nodes
3. Maximum average delay desired for short messages

In addition, the user may specify to the program a maximum cost that no network design will be allowed to exceed.

The output of the program is a sequence of low cost networks. Each network is identified by the following information:

1. Network topology
2. Cost per month
3. Maximum throughput
4. Estimated average traffic

5. Message cost per megabit at maximum throughput

6. Average message delay for short messages

Each acceptable network design also conforms to the standard that at least two nodes and/or links must fail before all communication paths between any pair of nodes are disrupted.

APPROACH

The general design problem as stated above is similar to other network design problems for which computationally practical solutions have recently been obtained. These problems include the minimum cost design of survivable networks, [3] the minimum cost selection and interconnection of Telpaks in telephone networks, [4] the design of offshore natural gas pipeline networks, [5] and the classical Traveling Salesman problem. [6] These problems have long resisted exact solution; however, recent work on approximate methods has been extremely successful and has led to efficient methods of finding low cost solutions in practical computation times.

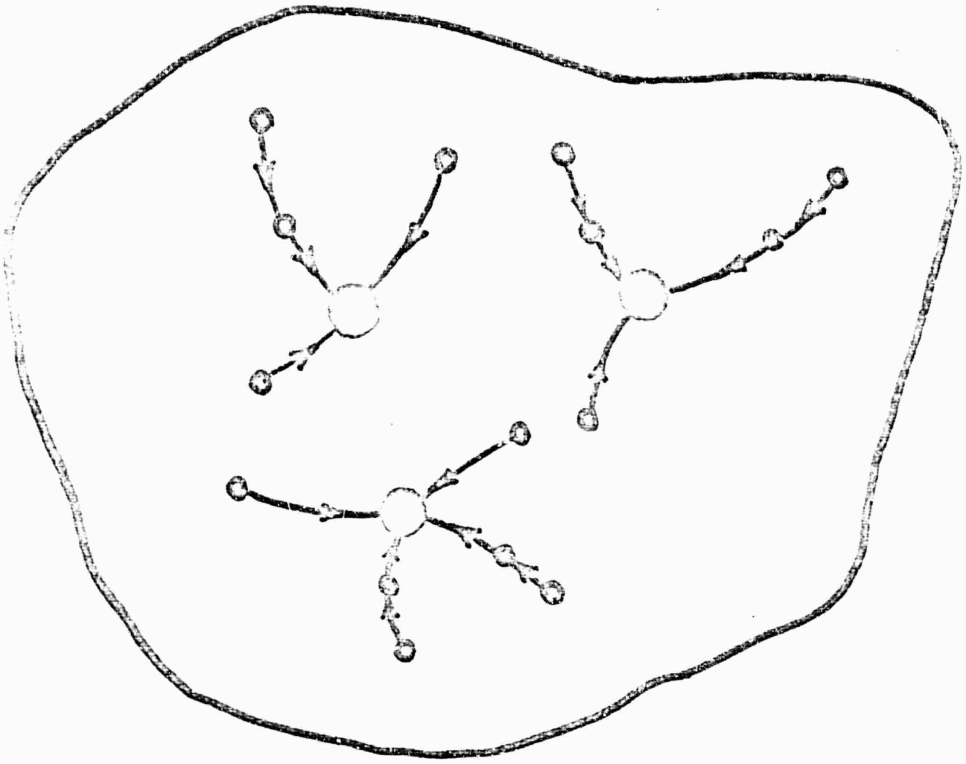
The Design Philosophy

By a "feasible" solution, we mean one which satisfies all of the network constraints. By an "optimal" network, we mean the feasible network with the least possible cost. Our goal is to develop a method that can handle realistically large problems in a reasonable computation time and which can find feasible solutions with costs close to optimal.

The method to be used has two main parts called the starting routine and the optimizing routine. The starting routine generates a feasible solution. The optimizing routine then examines networks derived from this starting network by means of local transformations applied to the network topology. When a feasible network with lower cost is found, it is adopted as a new starting network and the process is continued. In this way, a feasible network is eventually reached whose cost can not be reduced by applying additional local transformations of the type being considered. Such a network is called a locally optimum network.

Once a locally optimum network is found, the entire procedure is repeated by again using the starting routine. The starting routine may incorporate suggestions made by a human designer. For example, the present tentative configurations for the ARPA Network have been used. Alternatively, if desired, the starting routine may generate feasible networks without such advice. At the present time, our starting routine is capable of generating about 100,000 low cost networks.

By finding local optima from different starting networks, a variety of solutions can be generated. Figure 1 shows a diagrammatic representation of the process. The space of feasible



DIAGRAMATIC REPRESENTATION OF THE

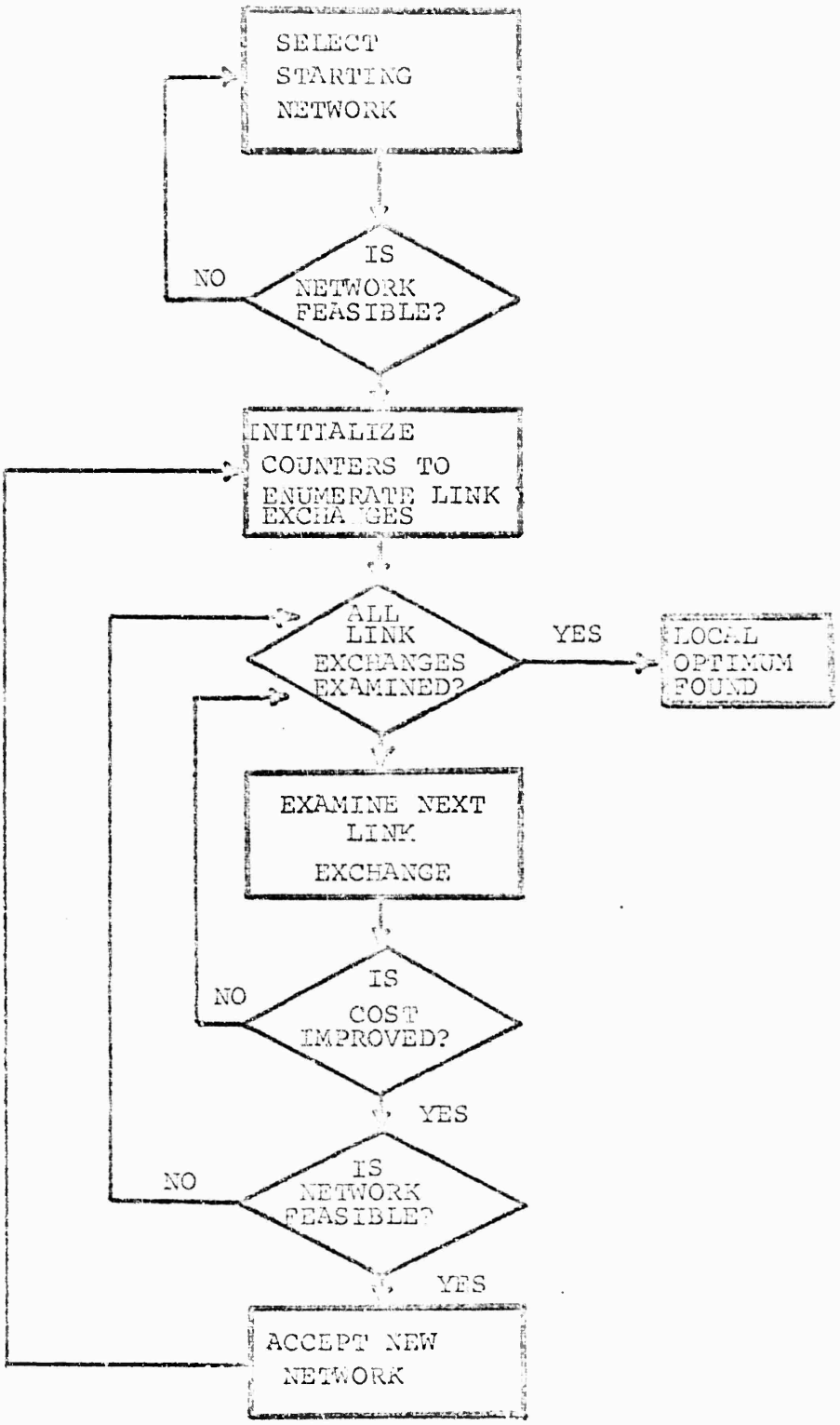
OPTIMIZATION PROCESS

FIGURE 1

solutions is represented by the area enclosed by the outer border of the figure; starting solutions are represented by light circles and local optima by dark circles. The practicality of the approach is based on the assumption that with a high probability some of the local optima found are close in cost to the global optimum. Naturally, this assumption is sensitive to the particular transformation used in the optimizing routine. A block diagram of the optimization procedure is shown in Figure 2.

Local Transformations

A local transformation on a network is generated by identifying a set of links, removing these links, and adding a new set to the network. The method of selection of the number and location of the links to be removed and added determines the usefulness of the transformation and its applicability to the problem in hand. For example, in the problem of economically designing offshore natural gas pipeline networks, dramatic cost reductions were achieved by removing and adding one link at a time.[5] On the other hand, in a problem of the minimum cost design of survivable networks, the most useful link exchange consisted of removing and adding two links at a time.[3] In general, it is not necessary that the same number of links be added and removed during each application of the transformation.



BLOCK DIAGRAM OF THE OVERALL PROCEDURE

FIGURE 2

DESIGN CONSTRAINTS

The preceding section has a given general approach for the design of low cost feasible networks. To implement this approach, a number of specific problems must be considered. These include:

1. The distribution of network traffic.
2. Network Route Selection.
3. Link capacity assignment.
4. Node and Link Time Delays.

Distribution of Traffic

At the present time, it is difficult to estimate the precise magnitude and distribution of the Host-to-Host traffic. However, one design goal is that the amount of flow that can be transmitted between nodes should not significantly vary with the locations of sender and receiver. Hence, two users several thousand miles apart should receive the same service as two users several hundred miles apart. A reasonable requirement is therefore that the network be designed so that it can accommodate equal traffic between all pairs of nodes. However, it is known that certain nodes have larger traffic requirements to and from the University of Illinois' Illiac IV than to other nodes. Con-

sequently, information of this type is incorporated into the model.

The magnitude of the network traffic is treated as variable. A "base" traffic requirement of 500·n bits per second (n is a positive real number) between all nodes is assumed. An additional 500·n bits per second is then added to and from the University of Illinois (node No. 9) and nodes 4, 5, 12, 18, 19, and 20. The base traffic is used to determine the flows in each link and the link capacities as discussed in the following sections. n is then increased until the average time delay exceeds .2 seconds. The average number of bits per second per node at average delay equal .2 seconds is taken as a measure of performance and the corresponding cost per bit is taken as a measure of efficiency of the network.

Route Selection

In order to avoid the prohibitively long computation times required to analyze dynamic routing strategies, a fixed routing procedure is used. This procedure is similar to the one which will be used in the operating network but it has the advantage that it can be readily incorporated into analysis procedures which do not depend on simulation.

The routing procedure is determined by the assumption that for each message a path which contains the fewest number of inter-

mediate* nodes from origin to destination is most desirable. Given a proposed network topology and traffic matrix, routes are determined as follows: For each i (i = 1, 2, ... , N = 20):

1. With node i as an initial node, use a labelling procedure [7] to generate all paths containing the fewest number of intermediate nodes, to all nodes which have non-zero traffic from node i. Such paths are called feasible paths.

2. If node i has non-zero traffic to node j (j = 1, 2, ... , N, j ≠ i) and the feasible paths from i to j contain more than seven nodes, the topology is considered infeasible.

3. Nodes are grouped as follows:

(a) All nodes connected to node i.

(b) All nodes connected to node i by a feasible path with one intermediate node.

(c) All nodes connected to node i by a feasible path with two intermediate nodes.

(d) - - - - - - - - - -

---

* A node j ≠ s,t is called an intermediate node with respect to a message with origin s and destination t if the path from s to t over which the message is transmitted contains node j.

(e) - - - - - - - - - - -

(f) All nodes connected to node i by a feasible path with five intermediate nodes.

Traffic is first routed from node i to any node j which is directly connected to i over link (i,j). Consequently, after this stage, some flows have been assigned to the network. Each node in group (b) is then considered. For any node j in this group, all feasible paths from i to j are examined, and the maximum flow thus far assigned in any link in each such path is found. All paths with the smallest maximum flow are then considered. The path whose total length is minimum is then selected and all traffic originating at i and destined for j is routed over this path.* All nodes in group (b) are treated in this manner. The same procedure is then applied to all nodes in group (c), (d), (e) and (f) in that order.

Capacity Assignment

Link capacities could be assigned prior to routing. Then after route selection, if the flow in any link exceeds its assigned capacity, the network would be considered infeasible. On the other hand, link capacities may be assigned after all traffic is routed;

---

* It is also possible to divide the traffic from i to j and send it over more than one feasible path, but for uniform traffic this is not an important factor.

we adopt this approach. The capacity of each link is chosen to be the least expensive option available from AT&T which satisfies the flow requirement. The line options which are presently being considered are: 50,000 bits/sec (bps), 108,000 bps, 230,400 bps, and 460,000 bps. Monthly link costs are the sum of a fixed terminal charge and a linear cost per mile. Thus, to satisfy a requirement of 85,000 bps, depending on the length of the link it is sometimes cheaper to use two 50,000 bps parallel links and sometimes cheaper to use a single 108,000 bps link.

The following line options and costs have been investigated:

Type	Speed	Cost Per Month
Full Group (303 data set)	50 KB	$ 850 + $ 4.20/mile
Full Group (304 data set)*	108 KB	$ 2400 + $ 4.20/mile
Telpak C	230.4 KB	$ 1300 + $ 21.00/mile
Telpak D	460 KB	$ 1300 + $ 60.00/mile

Link and Node Delays

Response time T is defined as the average time a message takes to make its way through the network from its origin to its destination. Short messages are considered to correspond to a single

---

* Not a standard AT&T offering.

packet which may be as long as 1008 bits or as short as few bits, plus the header. If T_i is the mean delay time for a packet passing through the i-th link, then $T = \frac{1}{r} \sum_{i=1}^{M} y_i \ T_i$, where r is the total IMP -to- IMP traffic rate, y_i is the average traffic rate in the ith link, and M is the total number of links. T_i can be approximated with the Pollaczak-Khinchin formula as:

$$T_i = \frac{1}{\mu C_i} \left[1 + \frac{y_i \ (1 + a^2)}{2 \ (\mu C_i - y_i)} \right]$$

where $1/\mu$ is the average packet length (in bits), C_i is the capacity of the ith link (in bits/second), a is the coefficient of variance for the packet length.

These parameters are evaluated as follows:

(1) r is the sum of all elements in the traffic matrix after each element has been adjusted to include headers, parity check and requests for next message (RFNM).

(2) y_i is determined by the routing strategy.

(3) In calculating $1/\mu$, we consider three kinds of packets: (a) packets generated by short messages and all other packets (except RFNM's) with length less than 1008 bits; (b) full length packets of 1008 bits belonging to long messages; (c) RFNM's.

It is assumed that the packets of part (a) are uniformly distributed with mean length equal to 560 bits. The packet length for part (b) is a constant equal to 1008 bits. The average packet length is then calculated by first estimating the average number of packets with 1008 bits. It is assumed that each long message consists of an average of 4 packets. In many of our computations, we assume that 80% of the messages are short. The number of RFNM packets can then be estimated. Finally, since the average length of each type of packet is known and the number of each type of packet has been estimated, the average packet length can be estimated.

(4) y_i is adjusted to include the increased traffic due to acknowledgments. C_i is then selected as already described.

(5) The larger the value of a, the larger the delay time. For the exponential distribution a = 1; for a constant, a = 0; and for many distributions $0 \leq a \leq 1$. Since it is reasonable to assume that the packet length distribution being considered is very close to the combination of an uniform distribution and a constant, the value of a should be less than one. To avoid underestimating T, a is set equal to one in all calculations.

The above analysis is based on the assumption that the number of available buffers is unlimited. When the traffic is low,

this assumption is very accurate. For high traffic, adjustments to account for the limitation of buffer space are necessary.

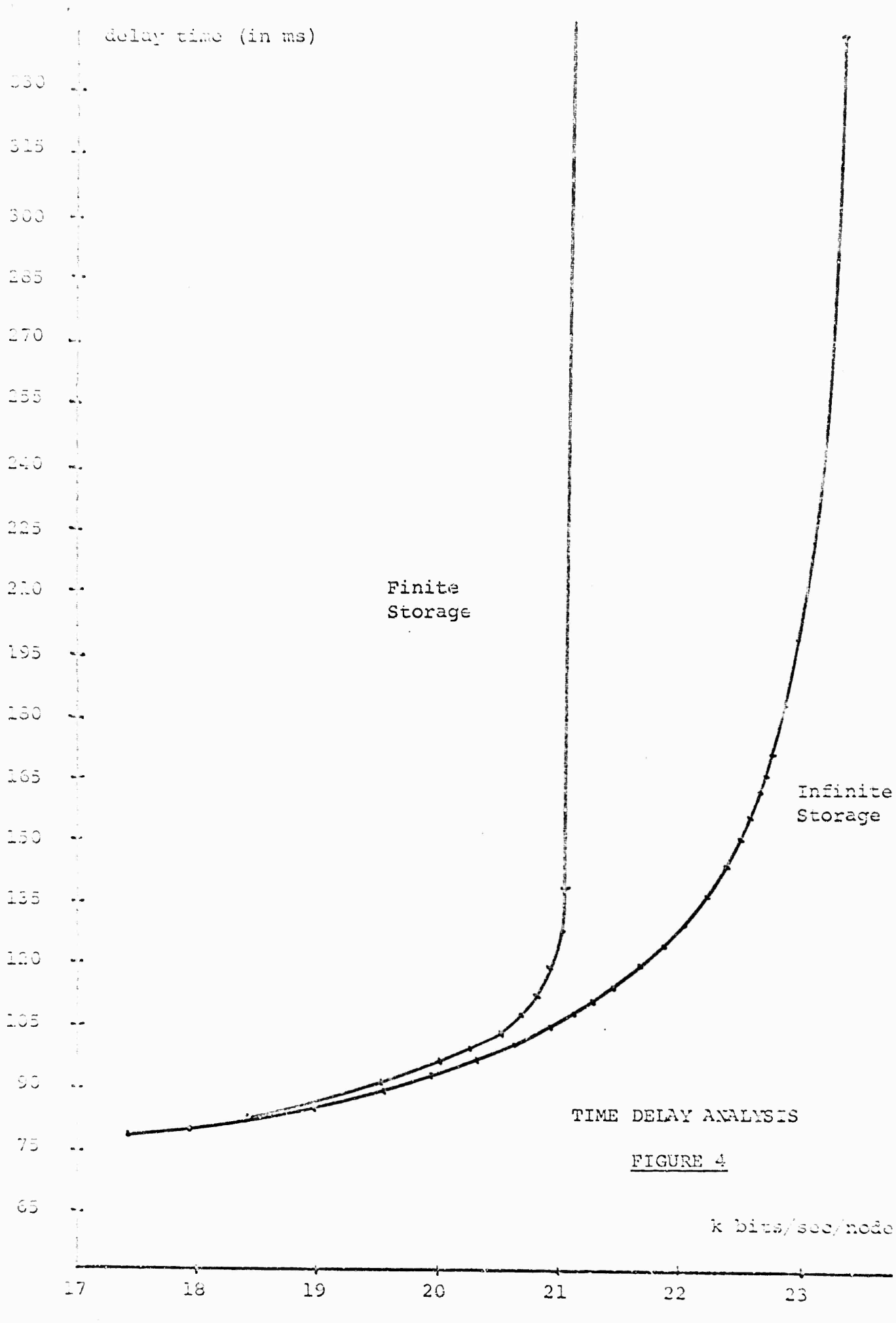
There are two roles for buffers in an IMP; one for reassembling messages destined for that IMP's Host and the other for store-and-forward traffic. At the present time, about one half of the IMP's core is used for the operating program. The remainder contains about 84 buffers each of which can store a single packet. Up to 2/3 of the buffers may be used for reassembly. Buffers not used for reassembly are available for store-and-forward traffic. When no buffer is available for reassembly, any arriving packet which requires reassembly but does not belong to any message in the process of reassembly will be discarded and no acknowledgment returned to the transmitting IMP. This packet must then be retransmitted, and the effective traffic in the link is therefore increased. In addition, each time a packet is retransmitted, its delay time is not only increased by the extra waiting and transmitting time, but also by the 100 ms time-out period. To account for these factors, an upper bound on the probability that no buffer is available is calculated for each IMP. The traffic between IMPS is then increased and extra delay time for the retransmitted packets is calculated. The increase in delay time is then averaged over all the packets.

NETWORK FOR TIME

DELAY ANALYSIS

FIGURE 3

When no buffer is available for store-and-forward traffic, all incoming links become inactive. Effectively, the average usable capacities of these links is lower than their actual capacities. The probability that no buffer is available for store-and-forward traffic is set equal to the average of an upper bound and a lower bound; the upper bound is calculated by assuming that the ratio of flow to capacity of each link into the IMP is equal to the maximum ratio for all links at that node while the lower bound is found by assuming that the ratio of flow to capacity for each link is equal to the minimum such ratio. Link capacities are then reduced to include this effect and the response time is then recalculated. An example of the effect of the above assumptions is shown in Figure 4. Figure 4 relates average time delay and throughput per node for the network shown in Figure 3. Two curves are shown. One is obtained by assuming that there are an infinite number of buffers at each node. The second curve is obtained by using the actual buffer limitations of the ARPA network.



TIME DELAY ANALYSIS

FIGURE 4

COMPUTATIONAL RESULTS

The computer program described above was employed to design many low cost networks under varying assumptions. In this section, we summarize the most significant of these results. Among the parameters that were varied in the designs were:

1. number and identity of nodes
2. link capacities
3. traffic levels

A maximum of twenty nodes as identified in the table below were considered. Layouts contained all or a subset of these nodes. The following cases will be discussed:

a. Twelve Node Networks containing Nodes 1-11, 14
b. Sixteen Node Networks containing Nodes 1-11, 13-17
c. Eighteen Node Networks containing Nodes 1-11, 13-18, 20

All nodes were constrained to have no more than 5 incident links and node 1, no more than 4 incident links.

TABLE 1

Node Number	Node Name	Node Location	
		LAT.	LONG.
1	UCLA	34 04	118 31
2	SRI	37 22	122 10
3	SB	34 30	119 45
4	UTAH	40 40	111 50
5	RAND	34 00	118 35
6	BBN	42 30	71 20
7	SDC	34 01	118 33
8	MAC	42 30	71 12
9	ILLINOIS	40 05	88 30
10	HARVARD	42 30	71 15
11	CARNEGIE-MELLON	40 30	79 50
12	LRL	37 33	122 44
13	BTL	40 45	74 15
14	LINCOLN LABS	42 35	71 20
15	CASE	41 30	81 45
16	STANFORD	37 18	122 10
17	MITRE	39 00	77 00
18	NCAR DENVER	39 30	105 00
19	PRINCETON	40 30	74 30
20	AFWS OMAHA	41 00	96 00

A major consideration is the effect of the 304 Data set on network cost and performance. This data set will allow a 50 Kilobit line to be driven at 108 Kilobits at no additional line cost. An additional terminal charge for this data set is required but this charge is independent of mileage and hence it can be an economical means of increasing the capacities of cross country lines. Since the capacities of the cross country lines often limit the overall capability of the network, it is to be expected that the 304 Data set option can enhance the network's operating performance. Networks were optimized with and without this option. Graphs of cost versus throughput for these cases are given below.

The effect of traffic levels and distribution upon performance was also examined. Traffic load is typically assumed to be at a uniform base level except for 100% additional traffic to and from node 9 and nodes 4, 5, 12, 18, 19, and 20. The base level of the traffic is then a design parameter. For a specified traffic matrix, flows are routed and capacities assigned to the links. The elements of the traffic matrix are then increased, thus increasing each link flow and the average time delay. The process is complete when the network saturates. At each step in the iteration a uniform percentage increase in the traffic matrix takes place. The vast majority of all design experiments followed

this example. However, in a few cases special studies were made to determine the effect of high concentrations of traffic between nodes 9, 18, and 19. These studies indicated that a "normally loaded" network (10 Kilobits/sec/Node) could accommodate this additional traffic without a substantial increase in cost if 108 Kilobit lines can be used.

Finally, the effect of using lines leased as of September 30, 1968 in the designs was examined. As of September, twelve lines connecting nine nodes had been ordered from AT&T. As part of our study, it was necessary to determine whether the use of these lines in the optimized networks would significantly effect the operating characteristics and economies when compared with the case when any lines could be used. Therefore, two sets of optimizations were performed; in one, the lines indicated in Table 2 were constrained to appear in all network designs; in the other, there were no such constraints. It was found that networks containing these lines could be designed which were as economical as the best networks found without these constraints.

TABLE 2

LEASED LINES AS OF SEPTEMBER 30, 1969

(1,2), (1,3), (1,5)

(2,3), (2,4)

(3,7)

(4,7), (4,9)

(5,6), (5,7)

(6,8)

(8,9)

It is important to note that the costs given are the cost of leasing lines and do not include the cost of the IMPS. Also, it must be emphasized that the results to be presented are empirical. Hence, we do not claim that the observations we present are definitive, and indeed it may be necessary to revise them. However, we feel that the following results can provide a useful step towards a better understanding of the behavior of store-and-forward computer networks.

As an example of the studies performed, we will discuss the design of twelve node networks. This is the smallest operating network which can be expected to test adequately the design philosophy of the ARPA Network. The twelve nodes considered are the first twelve to be activated in the ARPA Installation schedule and are nodes

labeled 1-11 and 14. One of the first goals in our study was to design networks which would operate effectively as both twelve node systems and then as twenty node systems when later expanded. The networks designs can be represented on a scatter diagram. The coordinate of the horizontal axis of the diagram is cost in dollars per year. The coordinate of the vertical axis is the average throughput per node* in bits per second for a specified distribution of traffic. The graph shown in Figure 5 is drawn for a specified maximum average message time delay of .19 seconds for short messages. Each point in the graph corresponds to a network generated, evaluated, and optimized by the computer.

To interpret these results, consider any point P_1 corresponding to a network N_1. Draw a horizontal line starting at P_1 to the right of P_1 and a vertical line down from P_1. Any point say P_2 which falls within the quadrant defined by the two lines is said to be dominated by P_1, since in a sense, network N_1 is "better than" network N_2. Similarly N_1 is said to be a dominant network. That is, for the same delay N_1 provides at least as much throughput as N_2 at no higher cost. Horizontal and vertical lines can be drawn through certain points $P_1, \ldots, P_n$ so that all other points are dominated by at least one of these. $P_1, \ldots, P_n$ thus represent, in one sense, the best networks.

---

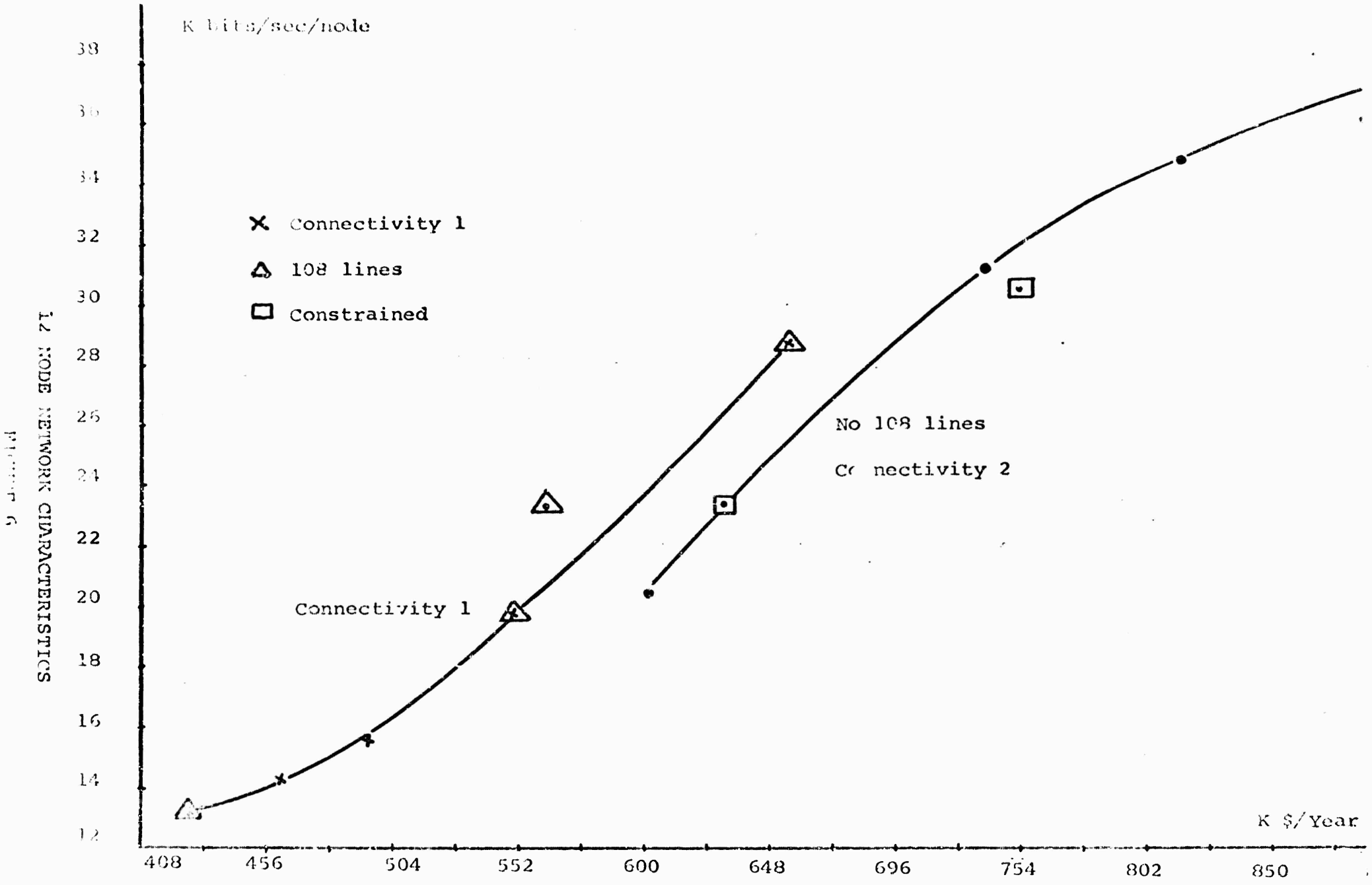
* Throughput is the average number of bits/second out of each node.

SCATTER DIAGRAM FOR 12 NODE NETWORKS
FIGURE 5

It should be noted that a network which is dominant for one time delay may not be dominant for another. Many networks with this property have been found in our studies. Furthermore, in some cases a network may be dominated but might still be preferable to the network which dominates it because of other factors such as the order of leasing lines and plans for future growth. As an example, P_1 is a dominant point and yet there are points which it dominates which are very close to it and might well be preferable.

Figure 6 indicates the cost-throughput characteristics of a number of dominant networks. In addition to the line cost per month and the average number of kilobits out of each node, we indicate whether the links given in Table 2 were constrained to be in the design. The presence of 304 data sets in the design, and the "connectivity" (i.e. the minimum number of nodes and or links whose failure will disconnect the network) are also indicated. Note that although many designs do not use 304 data sets, this option was available in all designs.

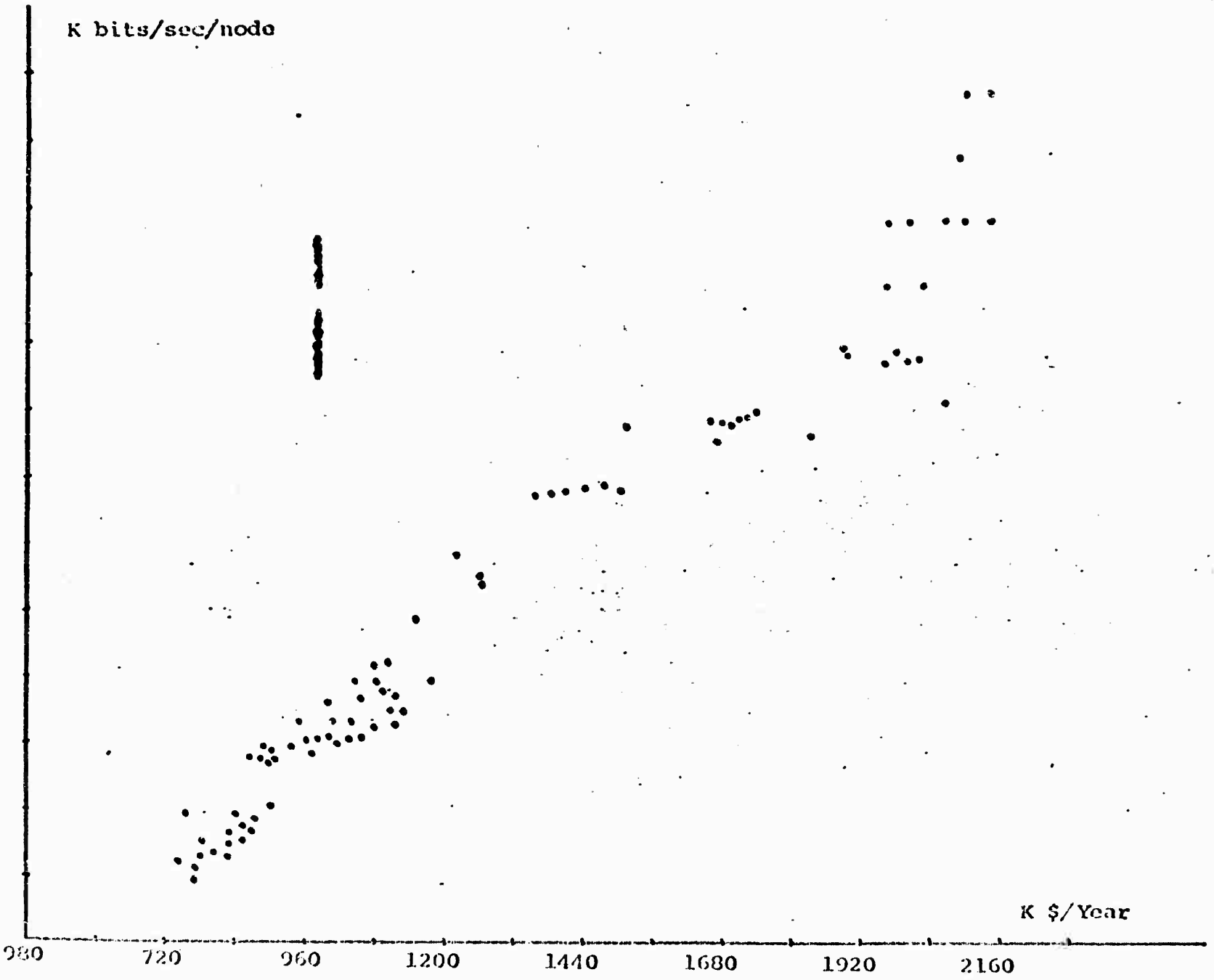
From Figure 6 it is clear that for rates below 29 kilobits/sec/node, significantly greater economies are obtainable with connectivity 1 networks than for connectivity 2 networks. This is because less lines need be used, data may be concentrated


K bits/sec/node
38
36
34
32
30
28
26
24
22
20
18
16
14
12
Connectivity 1
108 lines
Constrained
No 108 lines
Connectivity 2
Connectivity 1
K $/Year
408
456
504
552
600
648
696
754
802
850
17 NODE NETWORK CHARACTERISTICS

through fewer long lines, and 304 data sets used for these lines. By drawing a horizontal line between the connectivity 1 and connectivity 2 curves, the cost of the additional reliability of the connectivity 2 networks can be measured.

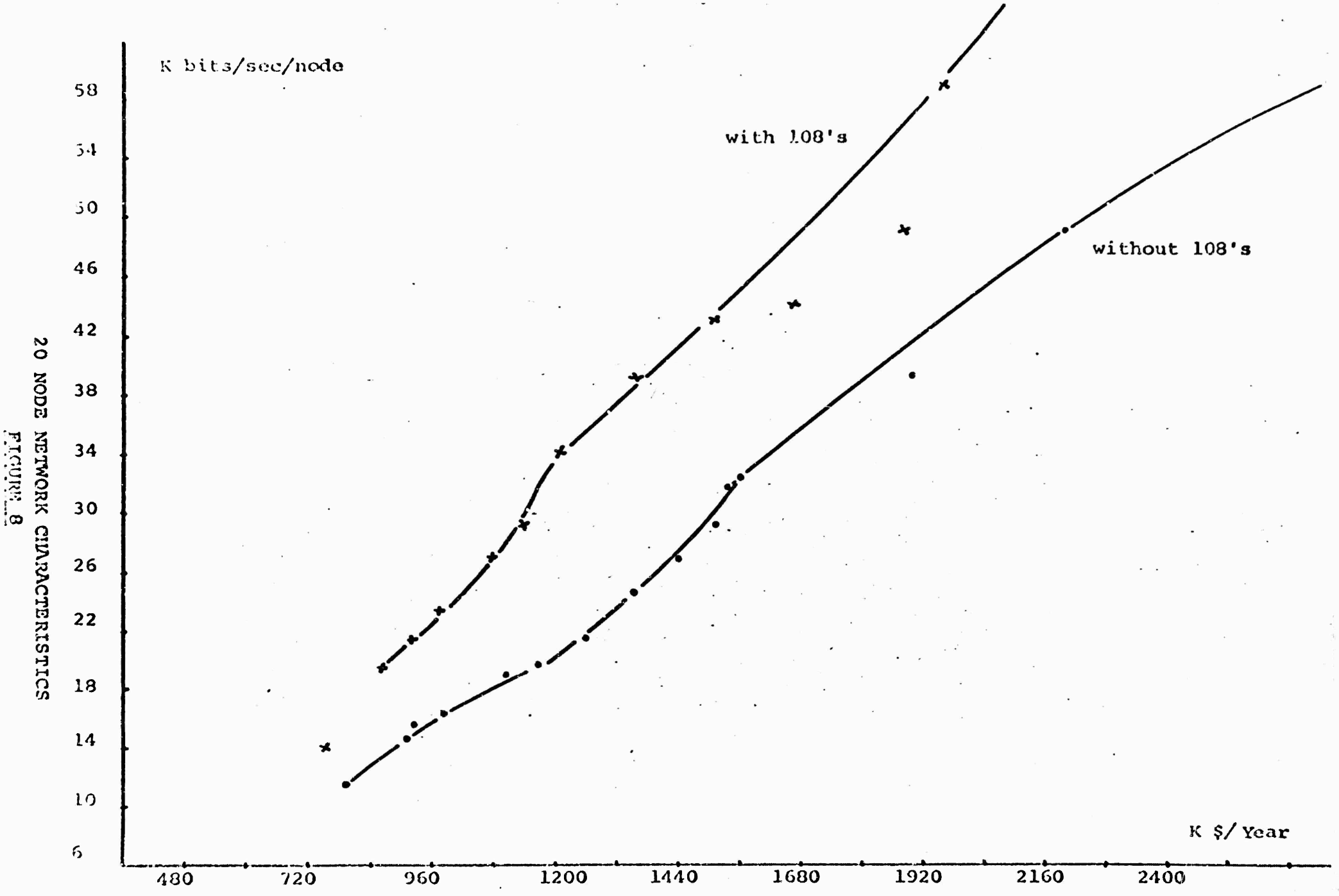
Figure 7 shows a scatter plot for 20 node networks designed with the 108 Kilobit/second 304 Data Set Option. Figure 8 indicates cost-throughput tradeoffs for 20 node networks with and without this option. Figure 9 presents this data in a different form - as a function of cost per megabit of transmitted information versus the required investment to achieve this cost. These costs were obtained by assuming that the network would be in use for 24 hours per day and hence for lesser utilized systems, the appropriate adjustments must be made.

Figure 10 summarizes the results of the network optimizations on 12, 16, 18, and 20 node networks without 108 kilobit lines. One immediate observation is that the node location and the traffic level are crucial factors in overall performance. In this figure, we plot total network cost against the total Host-to-Host traffic. Figure 11 shows total cost versus the average throughput per node for 16, 18, and 20 node networks with and without 108 kilobit lines.



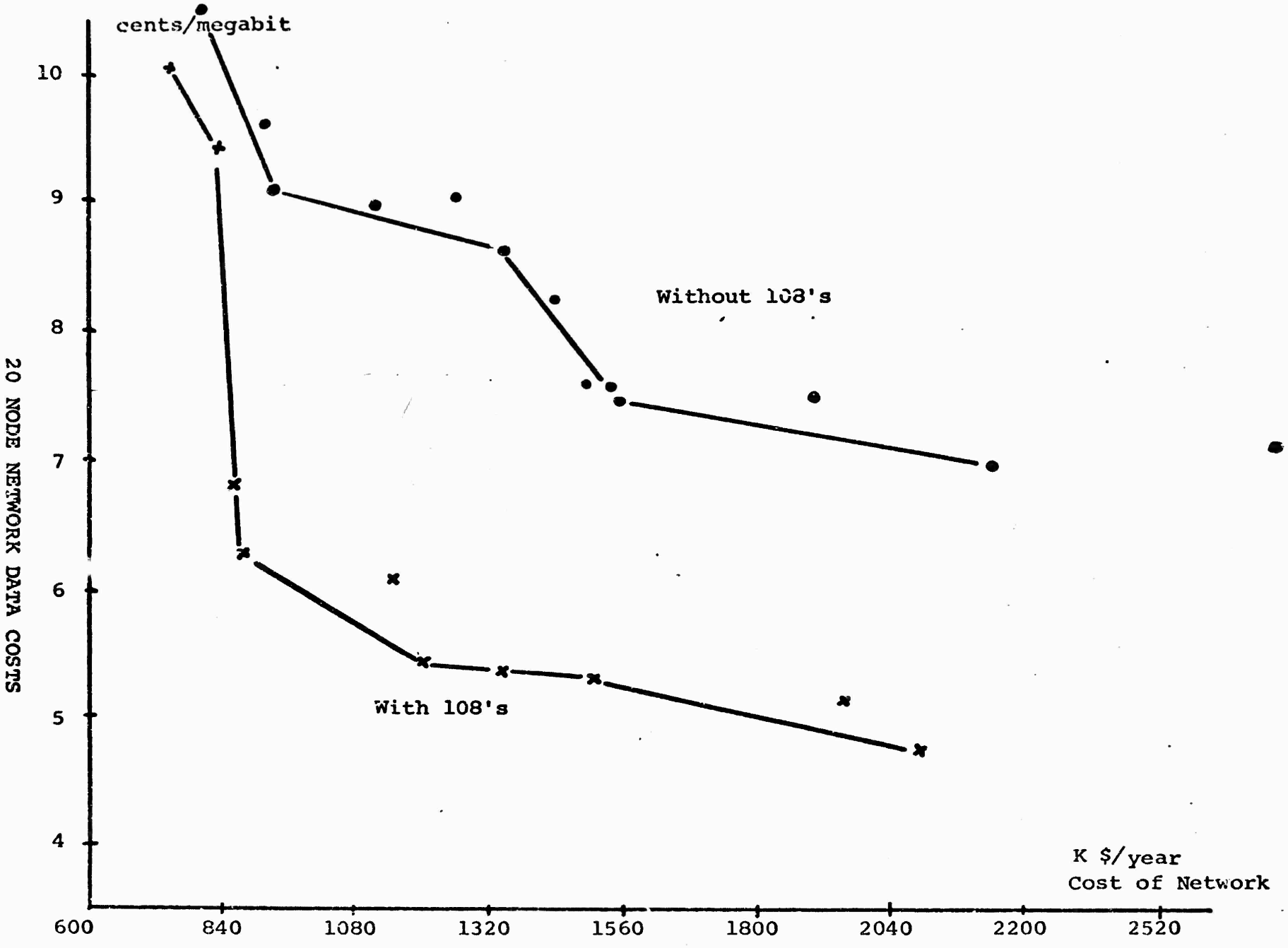
20 NODE NETWORK SCATTER DIAGRAM

FIGURE 7



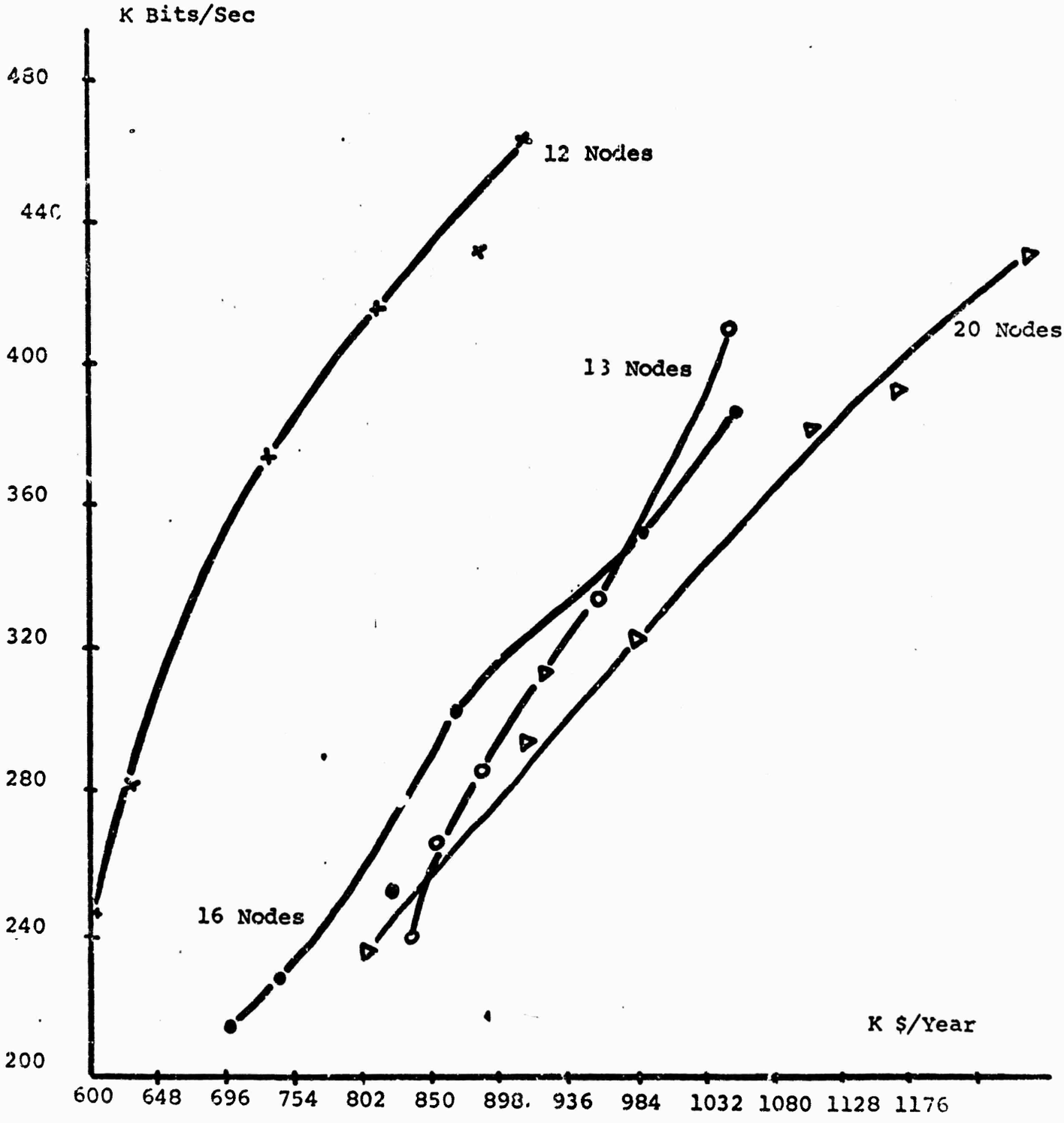
20 NODE NETWORK CHARACTERISTICS

FIGURE 8



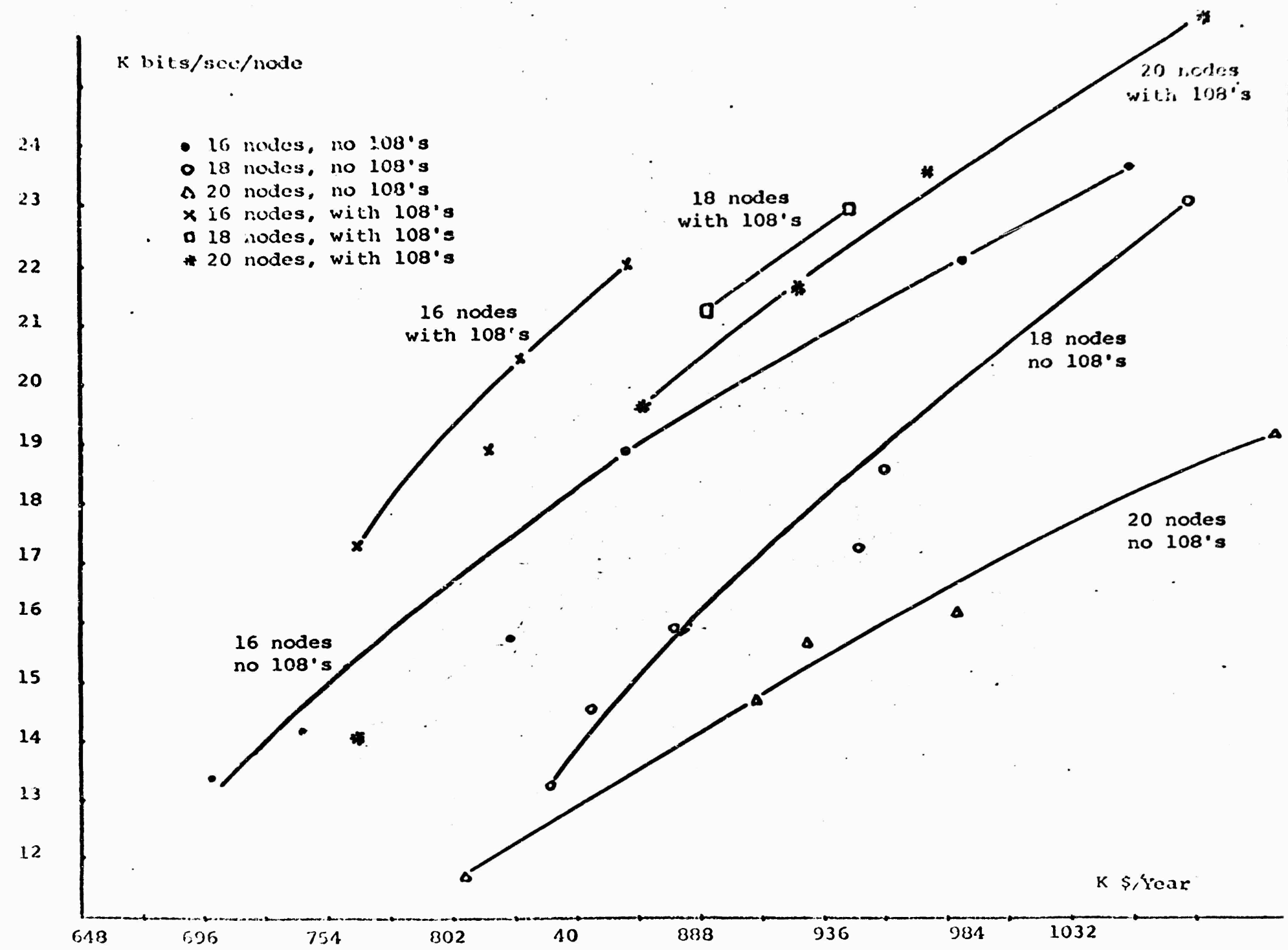
20 NODE NETWORK DATA COSTS

FIGURE 9



NETWORK CHARACTERISTICS WITHOUT 1.08 K BIT/SEC LINES

FIGURE 10

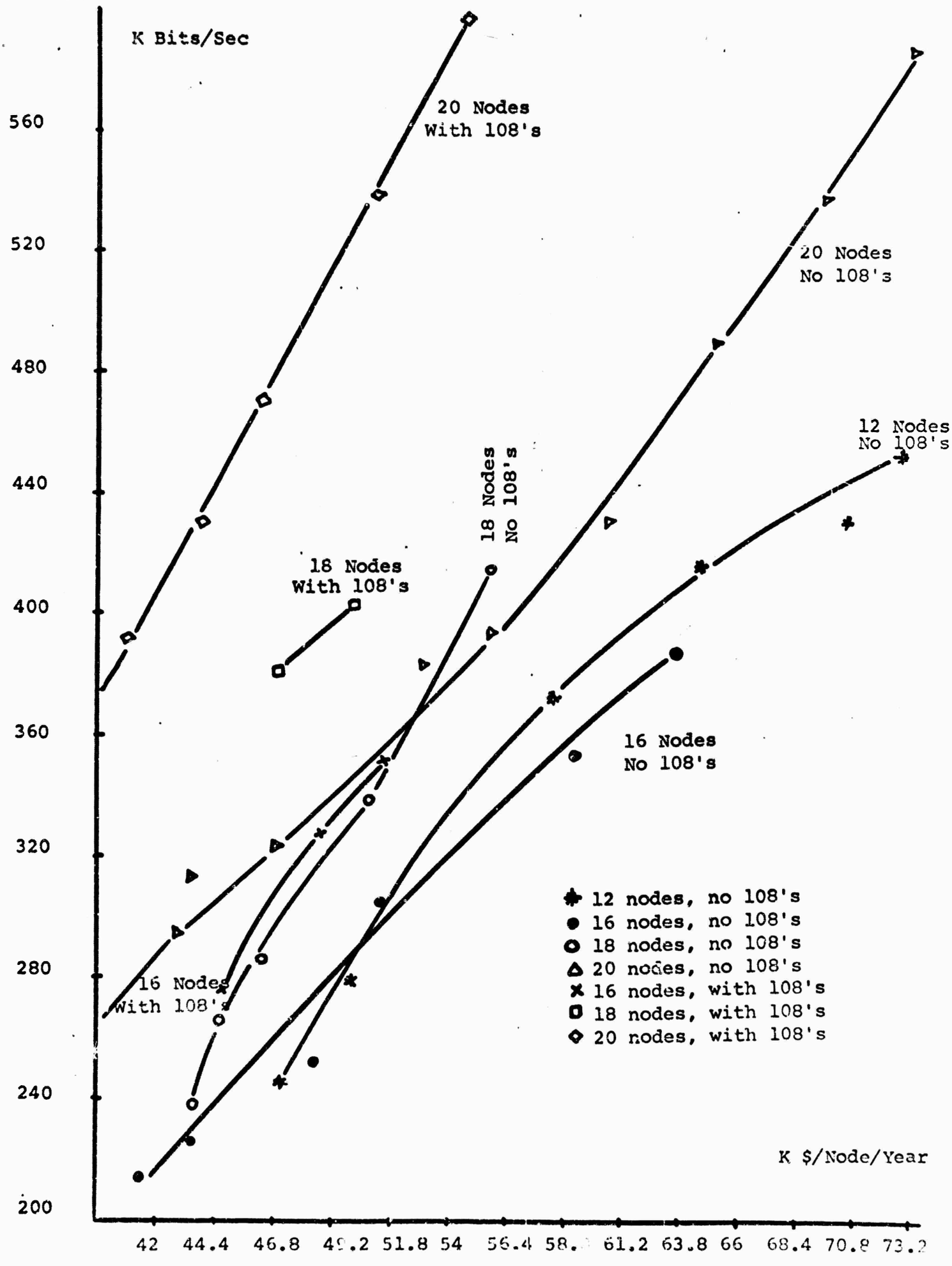


THROUGHPUT PER NODE VS. TOTAL NETWORK COST

FIGURE 11

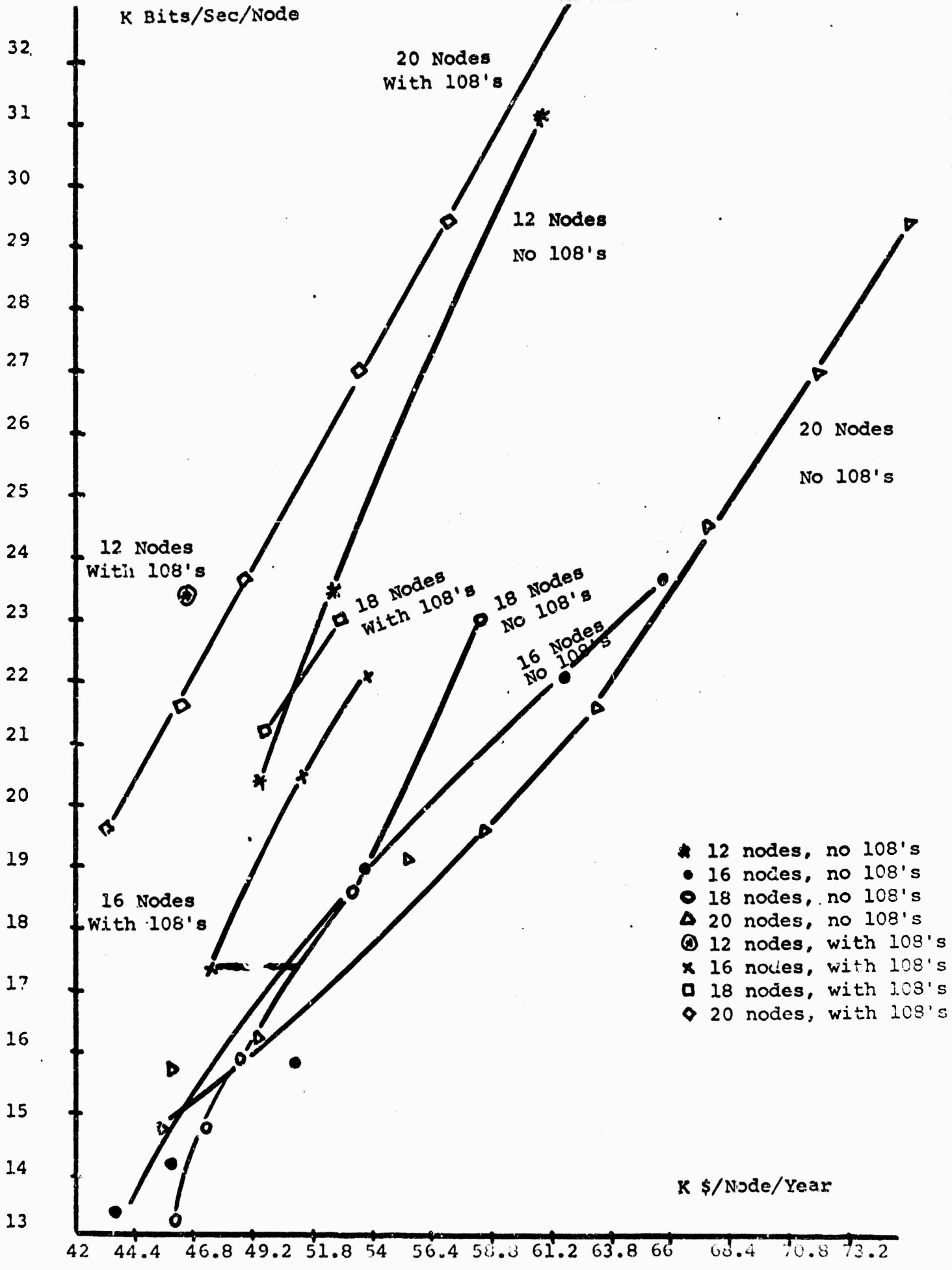
Figures 12 and 13 show the average cost per node versus the total Host-to-Host traffic and the average traffic per node, respectively. Cost-throughput characteristics are given for systems with and without 108 kilobit lines.

Finally, Figures 14 (a), (b) and (c) show typical computer designed networks. Figure 14 (a) shows a 12 node network using a 108 kilobit line. This was the only 12 node network (except for trivial modifications) which was found that had connectivity 2 and still used a 108 kilobit line. This network had the lowest cost - throughput ratio of all networks generated. Figure 14 (b) shows a very economical 18 node network designed without 108 kilobit lines. In addition to a low cost - throughput ratio, this network has two other desirable characteristics. First, it is economical as a 16 node network if nodes 18 and 20 are deleted (as well as lines (4,18), (18,20) and (20,15)) and a 50 kilobit/sec line is added from node 4 to 15. Second, the network's performance can be considerably enhanced by adding 50 Kilobit lines between nodes (3,7), (9,20), and (2,17). These additions result in a throughput increase of 7 Kilobits/sec node at an additional monthly cost of only $15,000. Figure 14 (c) shows a 20 node network, designed for low total cost at throughput levels projected for the ARPA Network. The 108 Kilobit option is allowed. To take advantage of this option the computer concentrates the



TOTAL TRAFFIC VS. COST PER NODE

FIGURE 12



THROUGHPUT PER NODE VS. COST PER NODE

FIGURE 13

$558,000/Year

23,400 Bits/Sec/Node

.16 Seconds Time Delay

HIGH THROUGHPUT 12 NODE NETWORK

FIGURE 14 (a)

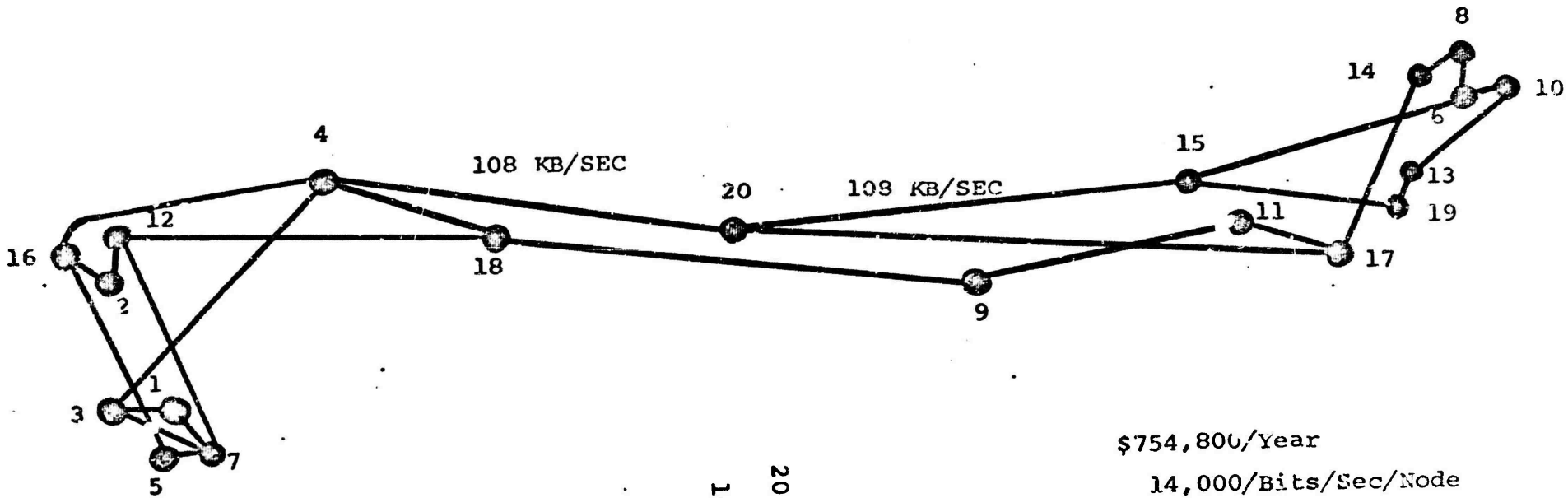
$883,000/Year

16,000 Bits/Sec/Node

.17 Seconds Time Delay

LOW COST 18 NODE NETWORK

FIGURE 14 (b)



$754,800/Year
14,000/Bits/Sec/Node
.182 Seconds Time Delay

20 NODE NETWORK WITH
108 K BIT/SEC LINES

FIGURE 14 (c)

traffic by transmitting the flow generated in the Los Angeles area to the San Francisco area where all cross country traffic is transmitted over high rate lines to the East Coast. A similar pattern occurs on the East-to-West Coast Traffic.

CENTRALIZED NETWORKS

Many computer networks consist of a set of remote sites connected to a central node. For example, most time sharing systems, computer reservation systems, accounting systems, etc., are of this type. In addition, many networks are hierarchal. That is, they are interconnections of centralized networks. The ARPA network seems to be evolving in this direction. At each node in the network a number of computers and terminals may eventually be connected. In addition, such systems must be considered in studying the economics of large computer networks. Major problems which arise in designing these systems are the layout and sizing of the connections between nodes. In order to solve these problems, the network designer is again faced with a discrete design problem which is intractable using existing integer programming methods for problems of practical size.

The objective is to select link locations and capacities so that the average time delay required to transmit a standard size message from any node to the central node does not exceed a specified number. This maximum allowable average delay time may, in some cases, vary from node to node. The design problem is then to find the least cost system which satisfies the time delay constraints for specified levels of traffic between nodes.

A strong case can be made for designing "tree" like centralized computer networks. That is, the nodes are connected

by the minimum number of possible links and there is exactly one transmission path between any pair of nodes. Although it is possible to construct situations in which trees are not optimal, they represent a reasonable class of networks for the layout problem. However, even if one reduces his range of designs to trees, the globally optimal network is usually impossible to find.

We now consider the optimal design of centralized computer networks. This study is a first step in our study of growth properties of store-and-forward networks. We describe a method to select globally least cost link capacities for a specified tree structure when maximum average delay times are specified for each node. We also give a heuristic method for finding low cost tree structures. These methods will be used to study the cost-time-delay-throughput characteristics of large hierarchal networks. The methods, which grew out of design studies for natural gas and irrigation systems, have been programmed and are capable of handling networks with several thousand nodes. In addition, they allow an arbitrary set of link capacities and cost structure and do not depend on the mathematical model used to calculate average time delay.

We consider the following network model.

(1) The network topology is a tree. The network contains N nodes numbered from 1 to N. A link between nodes i and j is denoted by the unordered pair (i,j). Node 1 is the central computer.

(2) Each link may be assigned one of a finite number of capacities $C_1, C_2, \ldots, C_k$. The capacity of link (j,k) is denoted by c(j,k) and for simplicity, each link is assumed to be fully duplex (not a necessary assumption).

(3) The cost of assigning capacity C to link (j,k) is an arbitrary function of various parameters such as distance, capacity, error rate, data set, and so on.

(4) The average time delay $\bar{t}(j,k)$ required to transmit b_i bits per second from node j to node k over link (j,k) can be expressed as $\bar{t}(j,k) = T(c(j,k), b_i \ldots\ldots) = T(c(j,k), \_)$ where _ represents all parameters other than c(j,k). The only property that we impose upon the function T(•) is the physically motivated one that $T(C_i, \_) > T(C_j, \_)$ if $C_j > C_i$. For example, we may use the time delay equations given earlier.

(5) Nodal time delays are insignificant. (This restriction can be removed but a complete treatment is lengthy).

(6) A traffix matrix $\underline{R} = [r_{i,j}]$ is specified where $r_{i,j}$ is the number of bits per second from node i to node j. All traffic from

node i to node j ($i \neq 1$, $j \neq 1$) must be routed through node 1. Thus, the network traffic can be described by two vectors

$$\underline{R}_1 = (r_{1,2}, r_{1,3}, \cdots, r_{1,N})$$

and

$$\underline{R}_2 = (r_{2,1}, r_{3,1}, \cdots, r_{N,1})$$

where

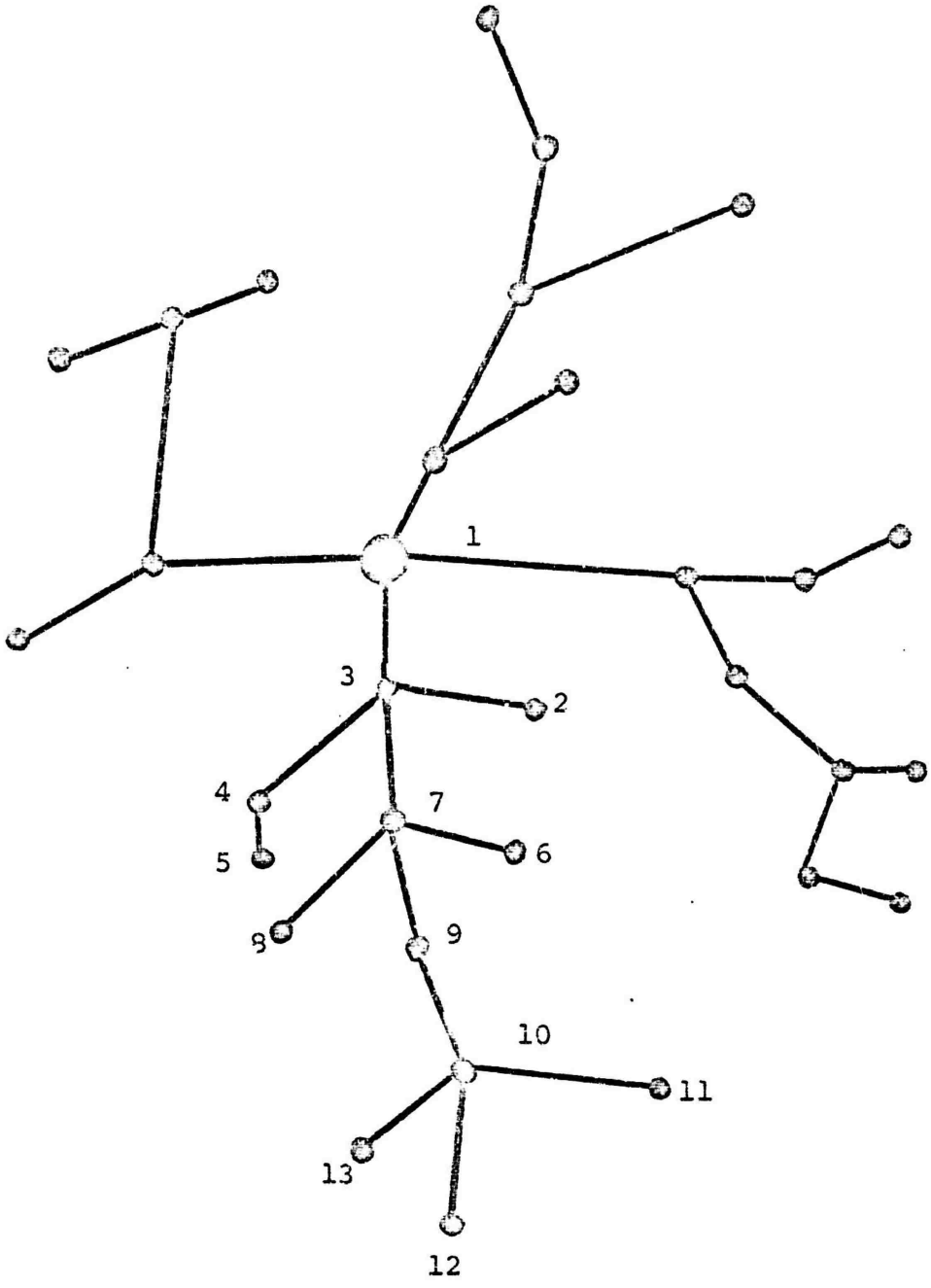
$$r_{1,i} = r_{1,i} + \sum_{\substack{j=2 \\ j \neq i}}^{N} r_{j,i}$$

$$r_{i,1} = r_{i,1} + \sum_{\substack{j=2 \\ j \neq i}}^{N} r_{i,j}$$

In a time sharing system with only one main computer, $r_{i,j}$ may equal zero when neither i nor j represents the central node 1. On the other hand, all of the off-diagonal entries in R may be non-zero for a computer-communication network.

Consider a fixed topological structure G such as the one shown in Figure 15. Capacities are to be assigned with minimal cost to this tree so that the maximum average time delay for transmission from any node to node 1 does not exceed t_{max}. A tree has the property that there is exactly one path between each pair of nodes. Consequently, given G, $\underline{R}_1$, and $\underline{R}_2$, the flows in each network link are uniquely determined. Let d_i, the "degree" of node i, be the number of links incident at i. A node j is said to be "pendant" if $d_j = 1$. Let t_i be the average time required to send a message from node i to node 1. With these definitions, it is easy to see that $\text{Max } \bar{t}_i \leq t_{max}$ if and only if $\text{Max}_{i:d_i=1} \bar{t}_i \leq t_{max}$. That is, in order to guarantee that t_{max} is not exceeded for transmission from any node in the network, we need only guarantee that this is true for pendant nodes. This property is used crucially in the algorithm to follow since limitations on network performance can be determined by considering only pendant nodes.

The first problem to be considered is given $\underline{R}_1$ and $\underline{R}_2$, find the least cost set of link capacities so that the maximum time delay required to transmit a message from any node to node 1 does not exceed a specified constant t_{max}. Choosing capacities for some of the links



CENTRALIZED NETWORK

FIGURE 15

and leaving the remaining capacities unspecified will be called a partial assignment. Methods will now be given to recognize partial assignments which cannot be in the optimal assignment. These partial assignments are discarded without discarding any partial assignment which might be in an optimal assignment.

Associate with each link two lists, called COST and DELAY. The i^{th} component of COST is the cost associated with the i^{th} smallest capacity choice. The i^{th} component of DELAY is the time delay for that link arising from a choice of the i^{th} smallest capacity. The values of the elements of COST are in increasing order and those of DELAY in decreasing order. The two lists taken together will be called a link array. Two techniques will now be given which, when used together on a given tree, can efficiently process these lists to obtain the optimal capacity assignment. These techniques were discovered by D. Kleitman and were first applied to the design of offshore natural gas pipeline networks. Variations and extensions of the algorithm have since been developed by the authors and applied to the design of Cable Television Systems and large scale irrigation systems. The following discussion is adapted from reference 5.

The first technique is called the parallel merge. The procedure can be used on any set of links which directly connect pendant nodes to a common node. We will use links (11, 10), (12, 10)

and (13, 10) from the tree of Figure 15 to illustrate the procedure. Suppose that there are seven possible capacities for each link and that the arrays for these links are as follows:

(11,10) array:
DELAY: (120, 111, 92, 66, 54, 40, 31)
COST: (13, 17, 23, 29, 36, 45, 58)

(12, 10) array:
DELAY: (150, 139, 118, 87, 75, 70, 67)
COST: (6, 9, 14, 21, 30, 40, 56)

(13, 10) array:
DELAY: (94, 86, 80, 61, 55, 48, 32)
COST: (8, 12, 18, 26, 34, 43, 57)

where the delays are in miliseconds and costs in hundreds of dollars. A testing block is set up as follows:

	(11,10)	(12,10)	(13,10)
DELAY	120 ms	(150)	94
COST	13	6	8
INDEX	1	1	1

Each link is assigned a column in the testing block as indicated. If the index in a column is set to i, then the DELAY and COST entries in that column will be the i^{th} components of the list. Initially, the indices are set to 1 and the testing block is as shown above.

The procedure locates the largest entry in the DELAY row of the testing block. In our example, this occurs in the column 2 and the entry is shown circled. If the smallest capacity is chosen for (12, 10), the delays at nodes 11 and 13 can never exceed t_{max}. Thus, choosing other than the minimum capacities for these links when (12, 10) has the minimum capacity will increase the total cost of the links but cannot reduce the maximum time delay.

We now enter the circled DELAY entry and the sum of the COST entries of the testing block in a new array. This entry on the new array corresponds to the partial assignment of minimum capacities to (11, 10), (12, 10) and (13, 10) and is shown below.

DELAY: (150)

COST: (27)

Since no better choice of capacities for (11, 10) and (13, 10) is possible with (12, 10) at this capacity, we increase the index in the second column of the testing blocks which yields

	(11,10)	(12,10)	(13,10)
DELAY	120	(139)	94
COST	13	9	8
INDEX	1	2	1

testing block

In the updated testing block, the new maximum DELAY entry is still in the second column. This means that if (12,10) has the second smallest capacity, it still will not pay to have (11,10) or (13,10) at any capacities other than the smallest. We make a second entry in the new list as before to give

DELAY: (150, 139)

COST: (27, 30)

This new entry represents a partial assignment of the second smallest capacity to (12,10) and the smallest capacity to (11,10) and (13,10).

The process terminates when the largest entry of the DELAY row of the testing block occurs in a column whose index has been promoted to its maximum value, 7. Further promotion of the other indices would correspond to partial assignments of greater cost and no possible savings in maximum time delay.

Each entry in the final new array (see below) represents an assignment of capacities to the links (11,10), (12,10) and (13,10) Furthermore, no other partial assignments for these links need be considered. Note that the number of possible partial assignments for these three links is $7^3 = 343$. However, the parallel merge techniques will produce an array with at most 19 columns, one from the original testing block and one each from the testing blocks resulting from a maximum of 18 index promotions. The minimum

number of columns in a new list is 7.

DELAY (150, 139, 120, 118, 111, 94, 92, 87, 86, 80, 75, 70, 67)

COST (27, 30, 35, 39, 46, 52, 56, 62, 71, 77, 85, 95, 111)

The parallel merge produces an array whose entries correspond to partial assignments which are candidates for inclusion in the optimal assignment. The new array can be viewed as the DELAY and COST lists of an equivalent link which replaces those links whose arrays were merged. This equivalent link can be thought of as a link connected between node 10 and a node cons sting of a combination of nodes 11, 12, and 13. (Hence, the name parallel merge.) Note that the components of the equivalent COST and DELAY lists are respectively in increasing and decreasing order so that no reordering is required.

It now becomes desirable to combine the array of the equivalent link with the array of (10,9) to create a new equivalent array for (11,10), (12,10), (13,10) and (10,9). Again we wish to retain as few partial assignments as possible without eliminating any partial assignments which can possibly be in the optimal assignment. A technique for accomplishing this, called the serial merge, is described next.

The serial merge can be used on any two links incident to a common node of degree two if at least one of the two links is also incident to a pendant node. We use the equivalent array obtained above and the following (10,9) array to illustrate the serial merge:

(10,9) array:

DELAY (133, 124, 104, 78, 65, 51, 42)

COST (6, 10, 15, 23, 33, 43, 59)

We set up a testing block with 7 columns as follows:

	1	2	3	4	5	6	7
DELAY	283	274	254	228	215	201	192
COST	33	37	42	50	60	70	86
INDEX	1	1	1	1	1	1	1

The i^{th} column corresponds to the i^{th} smallest capacity choice for (10, 9) and an index equal to j in a column corresponds to the j^{th} partial assignment of (11,10), (12,10), and (13,10) in their equivalent array. The DELAY and COST entries in a column are the corresponding maximum delay and the sum of link costs that would result from such a partial assignment of the four links.

Initially, the indices are all set to 1. The testing

block given above therefore gives all the data for the partial assignments for every choice of capacity for (10,9) with the partial assignment of (11,10), (12,10) and (13,10) corresponding to the first component of the equivalent array. Thus, the DELAY entry in the i^{th} column of the initial testing block is the sum of the i^{th} DELAY component for (10,9) and the first DELAY component on the equivalent branch list. Similarly the i^{th} column COST entry is the sum of the i^{th} (10,9) COST component and the first equivalent link COST component.

We now locate the maximum entry in the DELAY row of the testing block. Initially, this will always occur in the first column. The DELAY and COST entries of this column become candidate components in another equivalent link array. We then increase the index in the first column to yield

	1	2	3	4	5	6	7
DELAY	272	274	254	228	215	201	192
COST	36	37	42	50	60	70	86
INDEX	2	1	1	1	1	1	1

The largest DELAY entry is now in the second column. The DELAY and COST entries in this column become the second component in the new

array. The updating of the testing block yields

1	2	3	4	5	6	7
272	263	254	228	215	201	192
36	40	42	50	60	70	86
1	2	1	1	1	1	1

The new array becomes

DELAY: (283, 274, 272)

COST: (33, 37, 36)

Each column in the new array corresponds to a partial assignment of (11,10), (12,10), (13,10) and (10,9). The DELAY row has its components in non-increasing order, but the last COST component in the array is smaller than the second COST component. The partial assignment corresponding to the third column is <u>always</u> preferable to the partial assignment corresponding to the second column, since the former has a lower link cost and cannot result in a greater maximum time delay. We therefore can eliminate the second column from further consideration. Our array therefore reduces to

DELAY: (283, 272)

COST: (33, 36)

In general, when a new column is added to the array, we eliminate all columns already on the list with COST components which are not smaller than the latest entry. Since after each change the COST vector components in the array will be in increasing order, the updating of the array is easy to implement.

As we proceed with the serial merge, each of the 13 columns of the equivalent link array will form a candidate with each of the 7 (10,9) array components. Thus, a total of 7(13) = 91 candidates must be processed. However, as we have seen, some of these candidates can be eliminated. For the example under consideration, only 31 of the 91 candidates are retained and these constitute an equivalent link array for (11,10), (12,10), (13,10) and (10,9). In general only a small fraction of the candidates in a serial merge will be retained. A greater percentage of the earlier and later candidates will generally be retained than those in the middle, so the power of the elimination procedure is not fully illustrated by the small example given.

Since the parallel and serial merge techniques can be applied to lists of both actual and equivalent links, the entire tree can be processed to yield a single equivalent link array. The capacity assignment in this array with the smallest cost is the optimal assignment for the entire tree.

The size of the intermediate and final lists produced by the parallel and serial merge techniques are of great importance. The maximum list size appears to grow approximately linearly as a function of the number of nodes, where the number of possible assignments grows exponentially as a function of the number of nodes. It takes a fraction of a second of computer time on a CDC 6600 to optimize a 25 node tree. In addition, problems with several hundred nodes have been run in a few seconds. It appears that with careful programming and the application of a number of "short cuts", networks with as many as 10,000 nodes can be handled within a few minutes of computer time.

The above paragraphs describe an optimal method to efficiently select link capacities for a specified tree. We now give a heuristic method for finding low cost configurations. In combination with Kleitman's capacity assignment algorithm, this method appears to produce optimal or near optimal results in all cases.

As before, we will say that a "feasible" network is one which satisfies all of the network constraints and an "optimal" network is a feasible network with the least possible cost. The design method uses a <u>starting routine</u> to generate a feasible starting network and an <u>optimizing routine</u> to examine networks

derived from the starting network by means of a "local" change in network topology. If a feasible network with lower cost is found, it is adopted as a new starting network and the process repeated. Eventually, a locally optimal feasible network is reached, and the entire procedure is repeated with a different starting network. The starting network may be presented as an input or generated by the computer.

For the problem under consideration, an effective local transformation is a special kind of elementary tree transformation [7]. It can be shown that any tree can be obtained from any other tree by a sequence of elementary tree transformations. The elementary transformation used is as follows: For a given tree, choose a node i and find the node i_1 closest to i but not already connected to i. Add link (i_1,i) to the tree and identify the circuit formed. Suppose that this circuit consists of links (i_1,i), (i,i_2), (i_2,i_3) , ... , (i_j,i_1). New trees are formed by deleting in turn links (i,i_2), (i_2,i_3) , ... , and finally (i_j,i_1). Each time a link is deleted, Kleitman's algorithm is applied to determine optimal link capacities and network cost. As each node i is scanned in turn; link additions from i to its d nearest neighbors are considered. Whenever L lower cost trees have been generated, the node scan is begun again. (For offshore pipeline design [5], d=3 and L=1). This procedure has proven to be extremely powerful

in finding low cost trees. Whenever the problem has been small enough to exhaustively find optimal trees, the method has converged to the optimum solution. In only one case has a situation been constructed in which the local transformation could not produce the known lowest cost network.

REFERENCES

1. L. Kleinrock, "Models for Computer Networks," Proceedings of the International Conference on Communications, pp. 21.9 - 21.16, June, 1969.

2. L. Kleinrock, "Analytic and Simulation Methods in Computer Network Design," Proceedings of the Spring Joint Computer Conference, AFIPS Press, 1970.

3. K. Steiglitz, P. Weiner, and D. Kleitman, "Design of Minimum Cost Survivable Networks," IEEE Transactions on Circuit Theory, 1970.

4. B. Rothfarb and M. Goldstein, unpublished work.

5. H. Frank, B. Rothfarb, D. Kleitman and K. Steiglitz, Design of Economical Offshore Natural Gas Pipeline Networks, Office of Emergency Preparedness Report No. R-1, Washington, D. C., January, 1969.

6. S. Lin, "Computer Solutions of the Traveling Salesman Problem," Bell System Tech. Journal, Vol. 44, No. 10, pp. 2245 - 2269; December, 1965.

7. H. Frank and I. T. Frisch, Communication, Transmission, and Transportation Networks, Addison-Wesley, 1971.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
NETWORK ANALYSIS CORPORATION

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
None

3. REPORT TITLE
Analysis and Optimization of Store-and-Forward Computer Networks, Semiannual Technical Summary

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Semiannual Technical Summary (15 October 1969 - 15 June 1970)

5. AUTHOR(S) (Last name, first name, initial)
FRANK, HOWARD

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
15 June 1970	62	7

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAHC15-70-C-0120	ARPA Semiannual Report No. 1
b. PROJECT NO. OD30	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
c. ARPA Order No. 1523	-
d.	

10. AVAILABILITY/LIMITATION NOTICES
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Advanced Research Projects Agency, Department of Defense

13. ABSTRACT

This report discussed the analysis and optimization of the ARPA Computer Network. The general design philosophy followed as well as the specific elements considered in the implementation of this philosophy are described. Relationships between traffic level, link capacities, and cost as a function of the number of nodes in the networks have been investigated. Extensive studies have been made for twelve, sixteen, eighteen, and twenty node networks where each node was a potential site for the ARPA Network. Results of these studies are summarized. Methods for optimizing the design of centralized networks have been discovered. These methods, which are described here, are presently being used to design large decentralized networks.

14. KEY WORDS

Network Analysis	Store-and-Forward Networks
Computer Networks	Topological Optimization

UNCLASSIFIED
Security Classification